| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SerialArray.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PREV CLASS   [**NEXT CLASS**](http://docs.google.com/javax/sql/rowset/serial/SerialBlob.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sql/rowset/serial/SerialArray.html)    [**NO FRAMES**](http://docs.google.com/SerialArray.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

## **javax.sql.rowset.serial**

Class SerialArray

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.sql.rowset.serial.SerialArray**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Array](http://docs.google.com/java/sql/Array.html)

public class **SerialArray**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Array](http://docs.google.com/java/sql/Array.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html)

A serialized version of an Array object, which is the mapping in the Java programming language of an SQL ARRAY value.

The SerialArray class provides a constructor for creating a SerialArray instance from an Array object, methods for getting the base type and the SQL name for the base type, and methods for copying all or part of a SerialArray object.

Note: In order for this class to function correctly, a connection to the data source must be available in order for the SQL Array object to be materialized (have all of its elements brought to the client server) if necessary. At this time, logical pointers to the data in the data source, such as locators, are not currently supported.

**See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#javax.sql.rowset.serial.SerialArray)

| **Constructor Summary** | |
| --- | --- |
| [**SerialArray**](http://docs.google.com/javax/sql/rowset/serial/SerialArray.html#SerialArray(java.sql.Array))([Array](http://docs.google.com/java/sql/Array.html) array)            Constructs a new SerialArray object from the given Array object. |
| [**SerialArray**](http://docs.google.com/javax/sql/rowset/serial/SerialArray.html#SerialArray(java.sql.Array,%20java.util.Map))([Array](http://docs.google.com/java/sql/Array.html) array, [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> map)            Constructs a new SerialArray object from the given Array object, using the given type map for the custom mapping of each element when the elements are SQL UDTs. |

| **Method Summary** | |
| --- | --- |
| void | [**free**](http://docs.google.com/javax/sql/rowset/serial/SerialArray.html#free())()            This method frees the Array object and releases the resources that it holds. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getArray**](http://docs.google.com/javax/sql/rowset/serial/SerialArray.html#getArray())()            Returns a new array that is a copy of this SerialArray object. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getArray**](http://docs.google.com/javax/sql/rowset/serial/SerialArray.html#getArray(long,%20int))(long index, int count)            Returns a new array that is a copy of a slice of this SerialArray object, starting with the element at the given index and containing the given number of consecutive elements. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getArray**](http://docs.google.com/javax/sql/rowset/serial/SerialArray.html#getArray(long,%20int,%20java.util.Map))(long index, int count, [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> map)            Returns a new array that is a copy of a slice of this SerialArray object, starting with the element at the given index and containing the given number of consecutive elements. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getArray**](http://docs.google.com/javax/sql/rowset/serial/SerialArray.html#getArray(java.util.Map))([Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> map)            Returns a new array that is a copy of this SerialArray object, using the given type map for the custom mapping of each element when the elements are SQL UDTs. |
| int | [**getBaseType**](http://docs.google.com/javax/sql/rowset/serial/SerialArray.html#getBaseType())()            Retrieves the SQL type of the elements in this SerialArray object. |
| [String](http://docs.google.com/java/lang/String.html) | [**getBaseTypeName**](http://docs.google.com/javax/sql/rowset/serial/SerialArray.html#getBaseTypeName())()            Retrieves the DBMS-specific type name for the elements in this SerialArray object. |
| [ResultSet](http://docs.google.com/java/sql/ResultSet.html) | [**getResultSet**](http://docs.google.com/javax/sql/rowset/serial/SerialArray.html#getResultSet())()            Retrieves a ResultSet object that contains all of the elements in the ARRAY value that this SerialArray object represents. |
| [ResultSet](http://docs.google.com/java/sql/ResultSet.html) | [**getResultSet**](http://docs.google.com/javax/sql/rowset/serial/SerialArray.html#getResultSet(long,%20int))(long index, int count)            Retrieves a ResultSet object holding the elements of the subarray that starts at index *index* and contains up to *count* successive elements. |
| [ResultSet](http://docs.google.com/java/sql/ResultSet.html) | [**getResultSet**](http://docs.google.com/javax/sql/rowset/serial/SerialArray.html#getResultSet(long,%20int,%20java.util.Map))(long index, int count, [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> map)            Retrieves a result set holding the elements of the subarray that starts at Retrieves a ResultSet object that contains a subarray of the elements in this SerialArray object, starting at index *index* and containing up to *count* successive elements. |
| [ResultSet](http://docs.google.com/java/sql/ResultSet.html) | [**getResultSet**](http://docs.google.com/javax/sql/rowset/serial/SerialArray.html#getResultSet(java.util.Map))([Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> map)            Retrieves a ResultSet object that contains all of the elements of the SQL ARRAY value represented by this SerialArray object. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### SerialArray

public **SerialArray**([Array](http://docs.google.com/java/sql/Array.html) array,  
 [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> map)  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html),  
 [SQLException](http://docs.google.com/java/sql/SQLException.html)

Constructs a new SerialArray object from the given Array object, using the given type map for the custom mapping of each element when the elements are SQL UDTs.

This method does custom mapping if the array elements are a UDT and the given type map has an entry for that UDT. Custom mapping is recursive, meaning that if, for instance, an element of an SQL structured type is an SQL structured type that itself has an element that is an SQL structured type, each structured type that has a custom mapping will be mapped according to the given type map.

The new SerialArray object contains the same elements as the Array object from which it is built, except when the base type is the SQL type STRUCT, ARRAY, BLOB, CLOB, DATALINK or JAVA\_OBJECT. In this case, each element in the new SerialArray object is the appropriate serialized form, that is, a SerialStruct, SerialArray, SerialBlob, SerialClob, SerialDatalink, or SerialJavaObject object.

Note: (1) The Array object from which a SerialArray object is created must have materialized the SQL ARRAY value's data on the client before it is passed to the constructor. Otherwise, the new SerialArray object will contain no data.

Note: (2) If the Array contains java.sql.Types.JAVA\_OBJECT types, the SerialJavaObject constructor is called where checks are made to ensure this object is serializable.

Note: (3) The Array object supplied to this constructor cannot return null for any Array.getArray() methods. SerialArray cannot serialize null array values.

**Parameters:**array - the Array object to be serializedmap - a java.util.Map object in which each entry consists of 1) a String object giving the fully qualified name of a UDT (an SQL structured type or distinct type) and 2) the Class object for the SQLData implementation that defines how the UDT is to be mapped. The *map* parameter does not have any effect for Blob, Clob, DATALINK, or JAVA\_OBJECT types. **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if an error occurs serializing the Array object [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or if the *array* or the *map* values are null

### SerialArray

public **SerialArray**([Array](http://docs.google.com/java/sql/Array.html) array)  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html),  
 [SQLException](http://docs.google.com/java/sql/SQLException.html)

Constructs a new SerialArray object from the given Array object.

This constructor does not do custom mapping. If the base type of the array is an SQL structured type and custom mapping is desired, the constructor SerialArray(Array array, Map map) should be used.

The new SerialArray object contains the same elements as the Array object from which it is built, except when the base type is the SQL type BLOB, CLOB, DATALINK or JAVA\_OBJECT. In this case, each element in the new SerialArray object is the appropriate serialized form, that is, a SerialBlob, SerialClob, SerialDatalink, or SerialJavaObject object.

Note: (1) The Array object from which a SerialArray object is created must have materialized the SQL ARRAY value's data on the client before it is passed to the constructor. Otherwise, the new SerialArray object will contain no data.

Note: (2) The Array object supplied to this constructor cannot return null for any Array.getArray() methods. SerialArray cannot serialize null array values.

**Parameters:**array - the Array object to be serialized **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if an error occurs serializing the Array object [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or the *array* parameter is null.

| **Method Detail** |
| --- |

### free

public void **free**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

This method frees the Array object and releases the resources that it holds. The object is invalid once the free method is called.

After free has been called, any attempt to invoke a method other than free will result in a SQLException being thrown. If free is called multiple times, the subsequent calls to free are treated as a no-op.

**Specified by:**[free](http://docs.google.com/java/sql/Array.html#free()) in interface [Array](http://docs.google.com/java/sql/Array.html) **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if an error occurs releasing the Array's resources [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### getArray

public [Object](http://docs.google.com/java/lang/Object.html) **getArray**()  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html)

Returns a new array that is a copy of this SerialArray object.

**Specified by:**[getArray](http://docs.google.com/java/sql/Array.html#getArray()) in interface [Array](http://docs.google.com/java/sql/Array.html) **Returns:**a copy of this SerialArray object as an Object in the Java programming language **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if an error occurs retrieving a copy of this SerialArray object

### getArray

public [Object](http://docs.google.com/java/lang/Object.html) **getArray**([Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> map)  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html)

Returns a new array that is a copy of this SerialArray object, using the given type map for the custom mapping of each element when the elements are SQL UDTs.

This method does custom mapping if the array elements are a UDT and the given type map has an entry for that UDT. Custom mapping is recursive, meaning that if, for instance, an element of an SQL structured type is an SQL structured type that itself has an element that is an SQL structured type, each structured type that has a custom mapping will be mapped according to the given type map.

**Specified by:**[getArray](http://docs.google.com/java/sql/Array.html#getArray(java.util.Map)) in interface [Array](http://docs.google.com/java/sql/Array.html) **Parameters:**map - a java.util.Map object in which each entry consists of 1) a String object giving the fully qualified name of a UDT and 2) the Class object for the SQLData implementation that defines how the UDT is to be mapped **Returns:**a copy of this SerialArray object as an Object in the Java programming language **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if an error occurs

### getArray

public [Object](http://docs.google.com/java/lang/Object.html) **getArray**(long index,  
 int count)  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html)

Returns a new array that is a copy of a slice of this SerialArray object, starting with the element at the given index and containing the given number of consecutive elements.

**Specified by:**[getArray](http://docs.google.com/java/sql/Array.html#getArray(long,%20int)) in interface [Array](http://docs.google.com/java/sql/Array.html) **Parameters:**index - the index into this SerialArray object of the first element to be copied; the index of the first element is 0count - the number of consecutive elements to be copied, starting at the given index **Returns:**a copy of the designated elements in this SerialArray object as an Object in the Java programming language **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if an error occurs

### getArray

public [Object](http://docs.google.com/java/lang/Object.html) **getArray**(long index,  
 int count,  
 [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> map)  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html)

Returns a new array that is a copy of a slice of this SerialArray object, starting with the element at the given index and containing the given number of consecutive elements.

This method does custom mapping if the array elements are a UDT and the given type map has an entry for that UDT. Custom mapping is recursive, meaning that if, for instance, an element of an SQL structured type is an SQL structured type that itself has an element that is an SQL structured type, each structured type that has a custom mapping will be mapped according to the given type map.

**Specified by:**[getArray](http://docs.google.com/java/sql/Array.html#getArray(long,%20int,%20java.util.Map)) in interface [Array](http://docs.google.com/java/sql/Array.html) **Parameters:**index - the index into this SerialArray object of the first element to be copied; the index of the first element in the array is 0count - the number of consecutive elements to be copied, starting at the given indexmap - a java.util.Map object in which each entry consists of 1) a String object giving the fully qualified name of a UDT and 2) the Class object for the SQLData implementation that defines how the UDT is to be mapped **Returns:**a copy of the designated elements in this SerialArray object as an Object in the Java programming language **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if an error occurs

### getBaseType

public int **getBaseType**()  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html)

Retrieves the SQL type of the elements in this SerialArray object. The int returned is one of the constants in the class java.sql.Types.

**Specified by:**[getBaseType](http://docs.google.com/java/sql/Array.html#getBaseType()) in interface [Array](http://docs.google.com/java/sql/Array.html) **Returns:**one of the constants in java.sql.Types, indicating the SQL type of the elements in this SerialArray object **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if an error occurs

### getBaseTypeName

public [String](http://docs.google.com/java/lang/String.html) **getBaseTypeName**()  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html)

Retrieves the DBMS-specific type name for the elements in this SerialArray object.

**Specified by:**[getBaseTypeName](http://docs.google.com/java/sql/Array.html#getBaseTypeName()) in interface [Array](http://docs.google.com/java/sql/Array.html) **Returns:**the SQL type name used by the DBMS for the base type of this SerialArray object **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if an error occurs

### getResultSet

public [ResultSet](http://docs.google.com/java/sql/ResultSet.html) **getResultSet**(long index,  
 int count)  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html)

Retrieves a ResultSet object holding the elements of the subarray that starts at index *index* and contains up to *count* successive elements. This method uses the connection's type map to map the elements of the array if the map contains an entry for the base type. Otherwise, the standard mapping is used.

**Specified by:**[getResultSet](http://docs.google.com/java/sql/Array.html#getResultSet(long,%20int)) in interface [Array](http://docs.google.com/java/sql/Array.html) **Parameters:**index - the index into this SerialArray object of the first element to be copied; the index of the first element in the array is 0count - the number of consecutive elements to be copied, starting at the given index **Returns:**a ResultSet object containing the designated elements in this SerialArray object, with a separate row for each element **Throws:** SerialException, - which in turn throws an UnsupportedOperationException, if this method is called [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html)

### getResultSet

public [ResultSet](http://docs.google.com/java/sql/ResultSet.html) **getResultSet**([Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> map)  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html)

Retrieves a ResultSet object that contains all of the elements of the SQL ARRAY value represented by this SerialArray object. This method uses the specified map for type map customizations unless the base type of the array does not match a user-defined type (UDT) in *map*, in which case it uses the standard mapping. This version of the method getResultSet uses either the given type map or the standard mapping; it never uses the type map associated with the connection.

**Specified by:**[getResultSet](http://docs.google.com/java/sql/Array.html#getResultSet(java.util.Map)) in interface [Array](http://docs.google.com/java/sql/Array.html) **Parameters:**map - a java.util.Map object in which each entry consists of 1) a String object giving the fully qualified name of a UDT and 2) the Class object for the SQLData implementation that defines how the UDT is to be mapped **Returns:**a ResultSet object containing all of the elements in this SerialArray object, with a separate row for each element **Throws:** SerialException, - which in turn throws an UnsupportedOperationException, if this method is called [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html)

### getResultSet

public [ResultSet](http://docs.google.com/java/sql/ResultSet.html) **getResultSet**()  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html)

Retrieves a ResultSet object that contains all of the elements in the ARRAY value that this SerialArray object represents. If appropriate, the elements of the array are mapped using the connection's type map; otherwise, the standard mapping is used.

**Specified by:**[getResultSet](http://docs.google.com/java/sql/Array.html#getResultSet()) in interface [Array](http://docs.google.com/java/sql/Array.html) **Returns:**a ResultSet object containing all of the elements in this SerialArray object, with a separate row for each element **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if called, which in turn throws an UnsupportedOperationException, if this method is called

### getResultSet

public [ResultSet](http://docs.google.com/java/sql/ResultSet.html) **getResultSet**(long index,  
 int count,  
 [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> map)  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html)

Retrieves a result set holding the elements of the subarray that starts at Retrieves a ResultSet object that contains a subarray of the elements in this SerialArray object, starting at index *index* and containing up to *count* successive elements. This method uses the specified map for type map customizations unless the base type of the array does not match a user-defined type (UDT) in *map*, in which case it uses the standard mapping. This version of the method getResultSet uses either the given type map or the standard mapping; it never uses the type map associated with the connection.

**Specified by:**[getResultSet](http://docs.google.com/java/sql/Array.html#getResultSet(long,%20int,%20java.util.Map)) in interface [Array](http://docs.google.com/java/sql/Array.html) **Parameters:**index - the index into this SerialArray object of the first element to be copied; the index of the first element in the array is 0count - the number of consecutive elements to be copied, starting at the given indexmap - a java.util.Map object in which each entry consists of 1) a String object giving the fully qualified name of a UDT and 2) the Class object for the SQLData implementation that defines how the UDT is to be mapped **Returns:**a ResultSet object containing the designated elements in this SerialArray object, with a separate row for each element **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if called, which in turn throws an UnsupportedOperationException

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SerialArray.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PREV CLASS   [**NEXT CLASS**](http://docs.google.com/javax/sql/rowset/serial/SerialBlob.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sql/rowset/serial/SerialArray.html)    [**NO FRAMES**](http://docs.google.com/SerialArray.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).